Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

Кафедра Систем Управления и Информатики

**Лабораторная работа №1**

**Вариант №1**

Выполнили:

Новиков Н. В.

Ходакова М. А.

Борисевич А. В.

Проверил: Мусаев А. А.

Санкт-Петербург,

2023

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc146569853)

[1. Программа бинарного поиска 4](#_Toc146569854)

[2. Программа для определения одногруппников 4](#_Toc146569855)

[2.2 Описание программы 4](#_Toc146569856)

[2.2 Работа алгоритма 5](#_Toc146569857)

[3. Граф по второму заданию 6](#_Toc146569858)

[СПИСОК ИСТОЧНИКОВ 7](#_Toc146569859)

[ЗАКЛЮЧЕНИЕ 8](#_Toc146569860)

# ВВЕДЕНИЕ

Цель лабораторной работы №1 состояла в знакомстве с графами. Для е достижения были поставлены следующие задачи:

1. Написать программу для бинарного поиска;
2. Создать программу, которая по определенным характеристикам будет угадывать одногруппников;
3. Составить граф на основе программы из 2 задачи.

# 1. Программа бинарного поиска

Создадим некоторый список чисел list (числа в нём должны быть отсортированы по возрастанию) и зададим значение переменной x (именно её и будет искать программа). Введём переменную k – счётчик количества шагов. Обозначим за переменные left и right левую и правую границы поиска соответственно. В начале работы их значения равны первому и последнему элементам списка list.

Используем цикл while для нахождения числа. Цикл работает до тех пор, пока левая граница не больше правой (left <= right). При входе в цикл счетчик количества шагов увеличивается на 1, т. к. шаг по нахождению будет выполнен далее в цикле.

Шаг 1. Определяем середину области поиска (переменная mid) как среднее арифметическое её границ. Если элемент списка с индексом mid является искомым, то выходим из цикла и выводим ответ. В остальных случаях программа переходит к выполнению шага 2.

Шаг 2. Возможные ситуации разделим на два подслучая:

* если элемент списка с индексом mid меньше искомого числа, то сдвигаем левую границу поиска, присваивая ей значение mid+1, т. к. искомое число находится правее середины списка и элемент mid не является искомым;
* если элемент с индексом mid оказался больше искомого x. то сдвигаем правую границу поиска, присваивая её значение mid-1, т. к. искомое число находится левее середины списка.

Описанные выше шаги повторяются до тех пор, пока не будет найдена искомая переменная x или пока цикл не рассмотрит все элементы массива. В конце программа выведет потребовавшееся ей для поиска количество шагов.

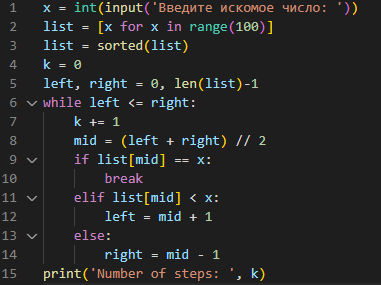


Рисунок 1 - Бинарный поиск

# 2. Программа для поиска одногруппников

## 2.2 Описание программы

Одной из целей лабораторной работы было создание программы, которая позволяет по определённым характеристикам угадать студента из группы. Программа была написана с помощью языка программирования Python без использования библиотек. Для описания всех одногруппников хватило 8-ми вопросов.

## 2.2 Работа алгоритма

Сначала создадим массив множеств, в каждое из которых будем помещать студентов, ответивших положительно на тот или иной вопрос. После создаем пустое множество persons, куда поместим всех возможных студентов.

Алгоритм проходится по всем студентам, добавляя их во множество всех возможных. Если студент ответил на i-ый вопрос положительно, то он добавляется во множество i из изначального массива (Рисунок 2). Так, в первом множестве будут находиться студенты, ответившие «да» на первый вопрос, во втором — студенты, ответившие «да» на второй вопрос и так далее.

После создания множеств программа проходится по всем вопросам в файле и ожидает ввода ответа от пользователя. В зависимости от ответа происходит происходит сужение круга возможных студентов. Рассмотрим два случая (Рисунок 3):

1. Если пользователь ответил на вопрос положительно. Программа пересекает множество всех возможных студентов со множеством студентов, ответивших на вопрос положительно. Очевидно, искомый студент входит в оба множества одновременно.
2. Если пользователь ответил на вопрос отрицательно. Из множества возможных студентов вычитается множество студентов, ответивших на тот же вопрос положительно. Искомый студент входит в первое множество, но не входит во второе.

Если в какой-то момент количество элементов во множестве всех возможных студентов окажется равно 1, то программа выведет имя загаданного студента. Случай, в котором после перебора всех студентов множество окажется пустым, будет означать, что студента с такими характеристиками нет в группе (Рисунок 4).

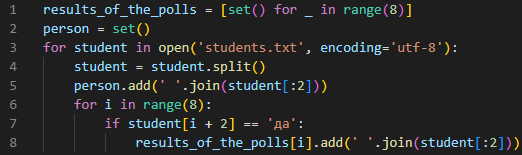


Рисунок 2 - Перебор всех студентов

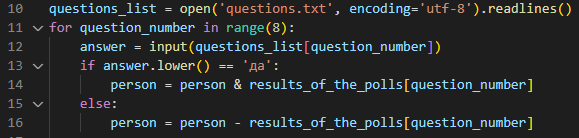


Рисунок 3 - работа со множеством студентов

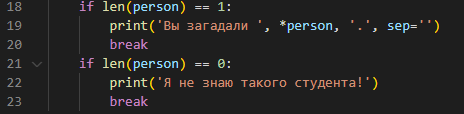


Рисунок 4 - Возможные исходы программы

# 3 Граф по второму заданию

Последним заданием лабораторной работы было создание графа по второму заданию. Этот граф является ориентированным деревом (Рисунок 5). Он не содержит циклов. В графе одна вершина имеет степень захода 0 (это вершина «Вопрос 1»), а все остальные вершины — степень захода 1. Помимо этого, все вершины в графе имеют степень исхода 2.

В данном графе каждый поворот влево (выбор левой дуги, идущей от вершины) означает ответ “нет” на заданный вопрос, а поворот вправо (выбор правой дуги) — “да”.

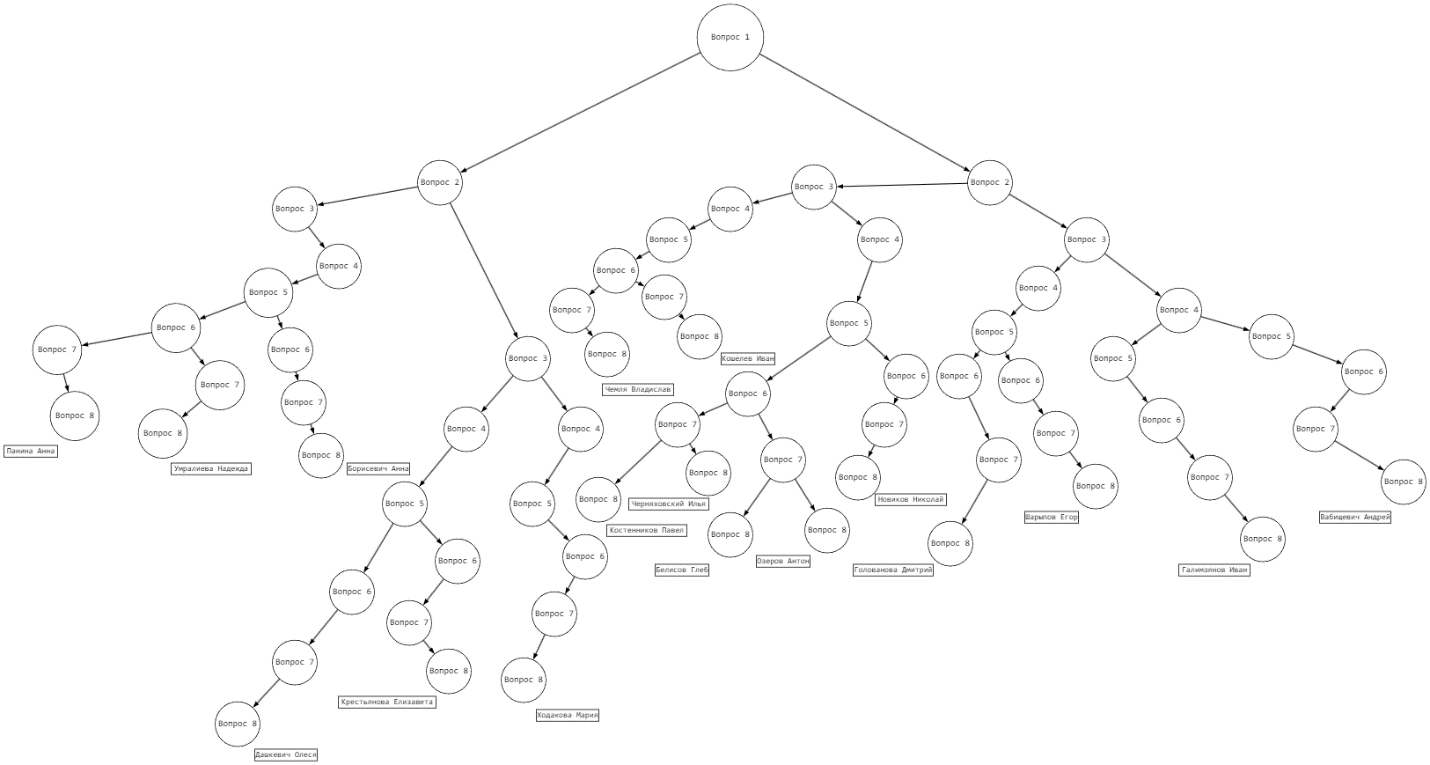


Рисунок 5 - Граф одногруппников

# ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы №1 были получены навыки работы с графами, а также с алгоритмом бинарного поиска, был написан алгоритм для поиска загаданного студента, основанный на работе со множествами.

# СПИСОК ИСТОЧНИКОВ

1. Википедия. Дерево (теория графов) [Электронный ресурс] [https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE\_(%D1%82%D0%B5%D0%BE%D1%80%D0%B8%D1%8F\_%D0%B3%D1%80%D0%B0%D1%84%D0%BE%D0%B2)](https://ru.wikipedia.org/wiki/Дерево_(теория_графов)) (Дата обращения 24.09.2023)